UNITED STATES PATENT AND TRADEMARK OFFICE

BEFORE THE BOARD OF PATENT APPEALS
AND INTERFERENCES

Ex parte Kevin Curtis Griffin, Scott Dennis Helt,
Michael James McDermott, Glen W. Nelson and Mark Philip Piazza

---

Appeal No. _____
Application No. 10/758,484

APPEAL BRIEF

Attorney Docket No. IBM/289 PATENT
Confirmation No. 6194

CERTIFICATE OF ELECTRONIC TRANSMISSION
I hereby certify that this correspondence for Application No. 10/758,484 is being electronically transmitted via EFS-WEB, on November 24, 2008.

/Scott A. Stinebruner/ November 24, 2008
Scott A. Stinebruner, Reg. No. 38,323 Date

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

Appellants:	Kevin Curtis Griffin et al.	Art Unit:	2186
Application No.:	10/758,484	Examiner:	Shane M. Thomas
Filed:	January 15, 2004		
For:	ASYNCHRONOUS HYBRID MIRRORING SYSTEM		

Mail Stop Appeal Brief - Patents
Commissioner for Patent
P.O. Box 1450
Alexandria, VA 22213-1450

APPEAL BRIEF

I. REAL PARTY IN INTEREST

This application is assigned to International Business Machines Corporation, of Armonk, New York.

II. RELATED APPEALS AND INTERFERENCES

There are no related appeals or interferences.

III. STATUS OF CLAIMS

Claims 1, 3, 4, 6-16 and 36 are pending in the Application, stand rejected, and are now on appeal. Claims 2, 5, 17-35, and 37 have been canceled.

IV. STATUS OF AMENDMENTS

There have been no amendments filed subsequent to the final rejection mailed September 15, 2008.

V. SUMMARY OF CLAIMED SUBJECT MATTER

Appellants' invention is generally directed towards methods for efficiently and reliably mirroring data of a primary system to a backup system. In one respect, a method consistent with embodiments of the invention provides a hybrid approach that enjoys benefits of both the fast processing of asynchronous mirroring and the data integrity of synchronous mirroring. More particularly, a number of update requests are organized into groups at both the primary and backup systems based on update requests from an application such that a first request from the application creates a first group of update requests and a second request from the application creates a second group of update requests. *See Application*, page 8 lines 2-13, page 11 line 25 to page 12 line 5, and page 13 lines 1-20, as well as FIG. 4 blocks 64-76. Although the first and second groups of update requests have an order dependency relative to each other, the update requests in each of the first and second groups of update requests are capable of being processed concurrently and without regard for order relative to one another. *See Application*, page 5 lines 2-11, page 5 lines 12-17, and page 8 lines 2-13, page 11 line 25 to page 12 lines 5, page 14 lines 16-24, and page 15 lines 18-22, as well as FIG. 4 blocks 76-82, FIG. 5 blocks 86-92, FIG. 6 blocks 102-144, and FIG. 7 blocks 152-166. Thus, the respective groups of update requests are completed sequentially to preserve sequential ordering of those groups. The updates of each group, however, are completed concurrently, or substantially at the same time and without regard to order. *See Application*, Abstract and page 5 lines 2-11, page 5 lines 12-17 and page 8 lines 2-13, page 11 line 25 to page 12 line 5, page 13 line 1 to page 14 line 24, page 14 line 25-31 and page 15 line 1 to page 17 line 2, as well as FIG. 4 blocks 76-82, FIG. 5 blocks 86-92, FIG. 6 blocks 102-144, and FIG. 7 blocks 152-166. This allows improved processing times with regard to the requests of each group.

Despite recent rapid advances in technology, computing systems partially or completely lose their ability to function properly during a crash or failure, losing valuable mission critical data. *See Application*, page 1 lines 16-27. Once practice used to protect critical data involves

data mirroring of the memory of a primary computer system to a backup computer system. Thus, write requests in the primary computer system are transmitted for execution to the backup computer system, and in the event that the primary computer system fails a user is switched to the backup computer system without a loss of significant amounts of data. *See Application*, page 1 line 28 to page 2 line 7.The prior art discloses two mirroring approaches for backing up data: the synchronous mirroring approach and the asynchronous mirroring approach. Synchronous mirroring involves updating data on the backup computer in the same order as the data updates on the primary computer system and preserves the order of the updates on the primary computer system. *See Application*, page 2 lines 8-15. However, synchronous mirroring often suffers from poor performance in that it takes a relatively long period of time for a particular update on the backup system to complete in proper order. *See Application*, page 2 line 25 to page 3 line 2. Thus, subsequent updates on the primary and backup systems are delayed and, over time, the ordered requirement of synchronous mirroring can result in unacceptable delays that adversely affect system performance. *See Application*, page 2 line 25 to page 3 line 2.

Asynchronous mirroring, on the other hand, provides improved performance over synchronous mirroring by issuing updates at the backup node without regard to order as the backup computer system attempts to process all received updates in parallel. *See Application*, page 3 lines 3-10. Thus, update requests are processed when received and while other requests are processing and the backup computer system puts the updates in proper sequential order after the application stops running. *See Application*, page 3 lines 3-10. However, this results in less integrity of the data on the backup computing system than asynchronous mirroring, as the primary computer system may crash and updates on the backup computer system may be processed out of order, rendering the data on the backup computer system out of order and unusable. *See Application,* page 3 lines 16-20.

Moreover, the inefficiency and unreliability of existing mirroring techniques becomes exacerbated in a clustered computer environment, as the nodes of a clustered computer environment process work in parallel. *See Application*, page 3 lines 21-30. As such, update requests for an application configured across the clustered computer environment may be completed simultaneously across a plurality of nodes, thus placing a larger burden on the clustered computing system to efficiently and accurately back up data from the plurality of nodes. *See Application*, page 3 lines 21-30.

Embodiments consistent with embodiments of the invention address these drawbacks by providing methods for updating data at a backup system that track updates to a primary system. Independent claim 1, for example, recites that, in response to receiving a first update request from an application, a first group is created that includes a first plurality of update requests which further includes the first update request. In response to receiving a second update request from the application prior to completing the first plurality of update requests, a second group is created that includes a second plurality of update requests which further includes the second update request. The first update request of the first plurality of update requests in the first group has an order dependency relative to the second update request of the second plurality of update requests in the second group, with the update requests in each of the first and second groups capable of being processed concurrently and without regard for order relative to one another. The method further includes concurrently completing the first plurality of update requests of the first group, and, after concurrently completing the first plurality of update requests, concurrently completing the second plurality of update requests of the second group.

Independent claim 14, for example, recites synchronously processing a plurality of groups of update request in that a first update request from an application in a first group of update requests from among the plurality of groups has an order dependency relative to a second update request from the application in a second group of update requests from among the plurality of groups, with the update requests in each group being capable of being processed concurrently and without regard for order relative to one another, and wherein receipt of the second update request prior to processing of the first update request initiates the creation of the second group of update requests. The method further comprises asynchronously processing the update requests in reach group.

Claims 1 and 14 are independent claims, while the rest of claims 3, 4, 6-13, 15, 16, and 36 are dependent claims. Claims 2, 5, 17-35, and 37 are canceled. For the convenience of the Board, each of the independent claims has been reproduced below and annotated with references to the specification and drawings to satisfy the requirement to concisely explain the claimed subject matter:

1. A method for updating data at a backup system (page 5 lines 2-11, page 8 lines 2-13, and page 14 line 25 to page 17 line 2, as well as FIG. 6 and FIG. 7) that tracks updates made to a

primary system (page 5 lines 2-11, page 8 lines 2-13, and page 12 line 22 to page 14 line 24, as well as FIG. 4 and FIG. 5), the method comprising:

in response to receiving a first update request from an application (page 12 lines 22-10 and page 14 lines 25-31, as well as FIG. 4 block 64 and FIG. 6 block 102), creating a first group (page 8 lines 2-13, page 11 line 25 to page 12 line 5, and page 13 lines 1-20, as well as FIG. 4 blocks 70-76) including a first plurality of update requests (page 8 lines 2-13, page 11 line 25 to page 12 line 5, page 12 line 22 to page 14 line 24, and page 14 line 25 to page 17 line 2, as well as FIG. 4 blocks 64-82 and FIG. 6 blocks 102-140), the first plurality of update requests including the first update request; (page 8 lines 2-13, page 11 line 25 to page 12 line 5, page 13 lines 1-20, and page 15 line 12 to page 17 line 2, as well as FIG. 4 blocks 70-76 and FIG. 5 blocks 116-144)

in response to receiving a second update request from the application prior to completing the first plurality of update requests (page 12 line 22 to page 13 line 20, page 14 lines 25-31, and page 15 line 12 to page 17 line 2, as well as FIG. 4 block 64 and 70-76, and FIG. 6 block 102 and 116-144), creating a second group (page 8 lines 2-13, page 11 line 25 to page 12 line 5, and page 13 lines 1-20, as well as FIG. 4 blocks 70-76) including a second plurality of update requests (page 8 lines 2-13, page 11 line 25 to page 12 line 5, page 12 line 22 to page 14 line 24, and page 14 line 25 to page 17 line 2, as well as FIG. 4 blocks 64-82 and FIG. 6 blocks 102-140), the second plurality of update requests including the second update request (page 8 lines 2-13, page 11 line 25 to page 12, line 5, page 13 lines 1-20, and page 15 line 12 to page 17 line 2, as well as FIG. 4 blocks 70-76 and FIG. 6 blocks 116-144), the first update request of the first plurality of update requests in the first group having an order dependency relative to the second update request of the second plurality of update requests in the second group (page 11 line 25 to page 12 line 5, page 13 lines 1-20, page 14 lines 16-24, and page 16 lines 6-29), with the update requests in each of the first and second groups capable of being processed concurrently and without regard for order relative to one another; (page 5 lines 2-11, page 5 lines 12-17, page 8 lines 2-13, page 11 line 25 to page 12 lines 5, page 14 lines 16-24, and page 15 lines 18-22, as well as FIG. 4 blocks 76-82, FIG. 5 blocks 86-92, FIG. 6 blocks 102-144, and FIG. 7 blocks 152-166) concurrently completing the first plurality of update requests of the first group; (page 5 lines 2-11, page 5 lines 12-17, page 8 lines 2-13, page 11 line 25 to page 12 line 5, page 13 line

1 to page 14 line 24, page 14 lines 25-31, and page 15 line 1 to page 17 line 2, as well as FIG. 4 blocks 76-82, FIG. 5 blocks 86-92, FIG. 6 blocks 102-144, and FIG. 7 blocks 152-166) and

after concurrently completing the first plurality of update requests, concurrently completing the second plurality of update requests of the second group. (page 5 lines 2-11, page 5 lines 12-17, page 8 lines 2-13, page 11 line 25 to page 12 line 5, page 13 line 1 to page 14 line 24, and page 14 line 25 to page 17 line 2, as well as FIG. 4 blocks 70-82, FIG. 5 blocks 86-92, FIG. 6 blocks 102-144, and FIG. 7 blocks 152-166).

14. A method for updating data at a backup system (page 5 lines 2-11, page 8 lines 2-13 and page 14 line 25 to page 17 line 2, as well as FIG. 6 and FIG. 7) that tracks updates made to a primary system (page 5 lines 2-11, page 8 lines 2-13, and page 12 line 22 to page 14 line 24, as well as FIG. 4 and FIG. 5) the method comprising:

synchronously processing a plurality of groups of update requests (page 5 lines 2-11, page 5 lines 12-17, page 8 lines 2-13, page 11 line 25 to page 12 line 5, page 13 line 1 to page 14 line 24, and page 14 line 25 to page 16 line 12, as well as FIG. 4 blocks 70-82, FIG. 6 block 102-144, and FIG. 7 blocks 152-166), a first update request from an application in a first group of update requests from among the plurality of groups having an order dependency relative to a second update request from the application in a second group of update requests from among the plurality of groups (page 11 line 25 to page 12 line 5, page 13 lines 1-20, page 14 lines 16-24 and page 15 line 12 to page 17 line 2), with the update requests in each group being capable of being processed concurrently and without regard for order relative to one another (page 5 lines 2-11, page 5 lines 12-17, page 8 lines 2-13, page 11 line 25 to page 12 line 5, page 12 line 22 to page 14 line 24, and page 15 lines 18-22, as well as FIG 4 blocks 64-74 and FIG. 5 blocks 86-92); and

asynchronously processing the update requests in each group. (page 5 lines 2-11, page 5 lines 12-17, page 8 lines 2-13, page 11 line 25 to page 12 line 5, page 13 line 1 to page 14 line 24, page 14 lines 25-31, and page 15 line 1 to page 17 line 2, as well as FIG. 4 blocks 76-82, FIG. 5 blocks 86-92, FIG. 6 blocks 102-144, and FIG. 7 blocks 152-166).

Other features will be discussed in greater detail in the arguments section below. In addition, it should be noted that, as none of the claims recite any means plus function or step plus function elements, no identification of such elements is required pursuant to 37 CFR

§41.37(c)(1)(v). Furthermore, there is no requirement in 37 CFR §41.37(c)(1)(v) to provide support for the subject matter in the separately argued dependent claims, as none of these claims recite means plus function or step plus function elements, and so no discussion of any of these claims is provided.

VI. GROUNDS OF REJECTION TO BE REVIEWED ON APPEAL

A. Claims 1, 6, 8, 10-15 and 36 are rejected under 35 U.S.C. § 103 (a) as being unpatentable over *Ohran* U.S. Patent No. 7,296,125 (hereinafter, *Ohran*) in view of Armangau et al. U.S. Patent 6,549,992 (hereinafter, *Armangau*) and in further view of *Ji* et al. U.S. Patent Application Publication No. 2004/0250029 (hereinafter *Ji*).

B. Claim 3 is rejected under 35 U.S.C. § 103 (a) as being unpatentable over *Ohran* in view of *Armangau*, *Ji*, and in further view of Yamagami U.S. Patent Application Publication No. 2004/0268067 (hereinafter, *Yamagami*).

C. Claim 4 is rejected under 35 U.S.C. § 103 (a) as being unpatentable over *Ohran* in view of *Armangau*, *Ji*, and in further view of Zait U.S. Patent Application Publication No. 2004/0210563 (hereinafter, *Zait*).

D. Claims 7 and 16 are rejected under 35 U.S.C. § 103 (a) as being unpatentable over *Ohran* in view of *Armangau*, *Ji*, and in further view of Kapoor et al. U.S. Patent Application Publication No. 2005/0021565 (hereinafter, *Kapoor*).

E. Claim 9 is rejected under 35 U.S.C. § 103 (a) as being unpatentable over *Ohran* in view of *Armangau*, *Ji*, and in further view of Golds et al. U.S. Patent No. 6,647,473 (hereinafter *Golds*).

VII. ARGUMENT

Appellants respectfully submit that the Examiner's rejections of claims 1, 3, 4, 6-16, and 36 are not supported on the record, are clear error, and should be reversed. All such claims have been rejected as being obvious over the prior art cited by the Examiner. Appellants respectfully

submit that, in the instant case, the Examiner has failed to establish a *prima facie* case of obviousness as to the aforementioned claims, and thus, the rejections thereof should be reversed.

Based on the Supreme Court's decision in KSR International Co. v. Teleflex Inc., 127 S. Ct. 1727, 1734, 82 USPQ2d 1385, 1382 (2007), a *prima facie* showing of obviousness still requires that the Examiner establish that the differences between a claimed invention and the prior art "are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art." 35 U.S.C. §103(a). Such a showing requires that all claimed features be disclosed or suggested by the prior art. Four factors generally control an obviousness inquiry: 1) the scope and content of the prior art; 2) the differences between the prior art and the claims; 3) the level of ordinary skill in the pertinent art; and 4) secondary considerations of non-obviousness, such as commercial success of products covered by the patent claims, a long felt but unresolved need for the invention, and failed attempts by others to make the invention. KSR, 127 S. Ct. at 1734 (quoting Graham v. John Deere Company, 383 U.S. 1, 17-18 (1966)) ("While the sequence of these questions might be reordered in any particular case, the [Graham] factors continue to define the inquiry that controls.").

Moreover, in KSR, the Court explained that "[o]ften, it will be necessary for a court to look to interrelated teachings of multiple patents; the effects of demands known to the design community or present in the marketplace; and the background knowledge possessed by a person having ordinary skill in the art, all in order to determine whether there was an apparent reason to combine the known elements in the fashion claimed by the patent at issue" and "[t]o facilitate review, this analysis should be made explicit." KSR, 127 S. Ct. at 1740-41 citing In re Kahn, 441 F.3d 977, 988, 78 USPQ2d 1329, 1336 (Fed. Cir. 2006) ("[R]ejections on obviousness grounds cannot be sustained by mere conclusory statements; instead, there must be some articulated reasoning with some rational underpinning to support the legal conclusion of obviousness."). But, not every combination is obvious "because inventions in most, if not all, instances rely upon building blocks long since uncovered, and claimed discoveries almost of necessity will be combinations of what, in some sense, is already known." KSR, 127 S. Ct. at 1741.

As a result, after KSR, while there is no rigid requirement for an explicit "teaching, suggestion or motivation" to combine references, there still must be some evidence of "a reason that would have prompted a person of ordinary skill in the relevant field to combine the elements in the way the claimed new invention does" in an obviousness determination. KSR, 127 S. Ct. at 1731.

Appellants respectfully submit that, in the instant case, the Examiner has failed to establish a *prima facie* case of obviousness as to independent claims 1 and 14, and as such, the rejections thereof should be reversed. As a matter of law, the rejections of dependent claims 3, 4, 6-13, 15, 16, and 36 necessarily must also be reversed. See, e.g., Hartness Int'l, Inc. v. Simplimatic Eng'g Co., 819 F.2d 1100, 1108, 2 USPQ2d 1826, 1831 (Fed. Cir. 1987) (dependent claims patentable if independent claims are patentable over the art). Although focus herein will primarily be on the independent claims, focus will be on the dependent claims where it is deemed necessary. Thus, Appellants' remarks in rebuttal to the Examiner's rejections are presented below, starting with relevant independent claims, and continuing with selected dependent claims that warrant separate mention. In some cases, specific discussions of particular claims are not made in the interests of streamlining this appeal. The omission of a discussion with respect to any particular claim, however, should not be interpreted as an acquiescence as to the merits of the Examiner's rejection of the claim, particularly with respect to claims reciting features that are addressed in connection with the rejections applied to other claims pending in the appeal.

A. Claims 1, 6, 8, 10-15, and 36 Are Patentable Over *Ohran*, *Armangau*, and In Further View of *Ji*

Independent Claim 1

Independent claim 1 generally recites a method for updating data at a backup system that tracks updates made to a primary system. The method comprises creating a first group including a first plurality of update requests in response to receiving a first update requests from an application. The first plurality of update requests includes that first update request. In response to receiving a second update request from the application prior to completing the first plurality of

update requests, a second group including a second plurality of update requests is created, the second plurality of update requests including the second update request. The first update request of the first plurality of update requests in the first group has an order dependency relative to the second update request of the second plurality of update requests in the second group. The update requests in each of the first and second groups are capable of being processed concurrently and without regard for order relative to one another. The method further comprises concurrently completing the first plurality of update requests of the first group, and, after concurrently completing the first plurality of update requests, concurrently completing the second plurality of update requests of the second group.

In rejecting claim 1, the Examiner relies on a combination of *Ohran, Armangau* and *Ji*. *Ohran* generally discloses snapshotting certain physical sectors of data at regular intervals, such as at T0, T1, T2, etc. *See Ohran*, Abstract and FIG. 2. As such, *Ohran* discloses snapshotting, at discrete times, only monitored physical sectors, and in some embodiments snapshotting only those monitored physical sectors that are changed. *See Ohran*, col. 2 lines 17-38. *Armangau*, on the other hand, merely discloses buffering and journaling of snapshots. *See Armangau*, Abstract. In particular, *Armangau* discloses that, in the event that backup data is streamed to a backup device faster than the backup data can be written on the backup device, the backup data is streamed to secondary storage and, when the backup device has caught up, that backup data is streamed from the secondary device to the backup device. *See Armangau*, col. 3 lines 29-36 and lines 48-56. In other embodiments, backup data is streamed to a computer, buffered, and then transferred to the backup device (*Armangau*, col. 3 line 57 to col. 4 line 11) and intermediate control logic throttles the stream of backup data (*Armangau*, col. 4 lines 18-35). *Ji* discloses collecting write transactions in batches based on batch size, then dispatching those batches to a secondary system. *See Ji*, Abstract and paragraph [0042], as well as FIG. 4.

Appellants respectfully submit, however, that *Ohran, Armangau* and *Ji* do not disclose or suggest a number of features recited in claim 1. In addition, Appellants submit that even if the cited references did arguably disclose each feature recited in claim 1, the cited references as a whole teach away from the claimed invention, and the modifications to those references

proposed by the Examiner are inharmonious with the teachings of those references. Each of these issues will be discussed separately below.

1. Creating Groups in Response to Update Requests Is Not Disclosed or Suggested

First, *Ohran*, *Armangau*, and *Ji*, either taken alone or in combination, fail to disclose or suggest creating a first group including a first plurality of update requests in response to receiving a first update request from an application and/or creating a second group including a second plurality of update requests in response to receiving a second update request from the application prior to completing the first plurality of update requests as recited in claim 1. In the most recent Office Action, the Examiner admits that *Ohran* fails to disclose or suggest creating a second group including a second plurality of update requests in response to receiving a second update request from the application prior to completing the first plurality of update requests. The Examiner is correct in this assertion, as *Ohran* fails to disclose or suggest creating any group in response to receiving any update request from the application. However, Appellants respectfully submit that *Ohran*, *Armangau*, and *Ji*, whether taken alone or in combination, fail to disclose or suggest creating any group in response to receiving any update request from the application, let alone creating the second group in response to receiving the second update request.

Ohran discloses a system that sends a record of monitored sectors as they exist at a particular time. If the sectors are subject to ten update requests between backup times, *Ohran* would copy the data of the sectors as they existed at a backup time and not the ten update requests directed to the sectors between backup times. An embodiment consistent with the claimed invention, however, would send the ten update requests directed to the record and, if the ten update requests come from the same application, would create ten respective groups in which to send those ten update requests. For example, at T1, T2, T3, and subsequent time periods, tracked changes of the original data are backed up to a snapshot copy. *See Ohran*, FIG. 2, as well as the discussion of FIG. 2 in col. 6 lines 18-60. These snapshots are not groups of "update requests" as recited in claim 1, nor are these snapshots created in response to receiving an update request from an application. Thus, *Ohran* explicitly discloses that snapshotting changed data occurs based on intervals of time and operates in a fundamentally different manner than as the

method recited in claim 1, which creates groups specifically in response to receiving update requests.

It may be that the Examiner focuses on the two teachings in *Ohran* that (1) an application or software element makes changes to the data and that (2) data is snapshotted. Thus, the Examiner may allege that *Ohran* discloses creating the first group in response to receiving the first update request. However, Appellants respectfully submit that the teachings of *Ohran* are still insufficient to disclose or suggest creating a group in response to receiving an update request. Rather, *Ohran* discloses a system that sends a record of monitored sectors as they exist at a particular time. Appellants respectfully submit that *Ohran* is wholly unconcerned with how data has been updated, and only concerned with the data at the time of the snapshot. As such, *Ohran* is necessarily unconcerned with the creation of groups of update requests, and understandably does not disclose or suggest creating a first group including a first plurality of update requests in response to receiving a first update request from an application and/or creating a second group including a second plurality of update requests in response to receiving a second update request from the application prior to completing the first plurality of update requests.

Moreover, *Armangau* and *Ji* fail to relieve the deficiencies of *Ohran*, as *Armangau* and *Ji*, whether taken alone or in combination, also fail to disclose or suggest creating a first group including a first plurality of update requests in response to receiving a first update request from an application and creating a second group including a second plurality of update requests in response to receiving a second update request from the application prior to completing the first plurality of update requests recited in claim 1. Rather, *Armangau* is directed towards buffering snapshotted data, and thus completely fails to disclose or suggest creating groups of update requests at all. Likewise, *Ji* is solely directed towards collecting write transactions in batches based on batch size. *See Ji*, paragraph [0023]. Significantly, the Examiner admits that *Ji* discloses creating batches of write transactions based on the size of the batch. See Final Office Action, page 2. As such, Appellants respectfully submit that although *Ji* does recite collecting write transactions in batches, *Ji* still fails to disclose or suggest creating a first group including a first plurality of update requests in response to receiving a first update request from an application and/or creating a second group including a second plurality of update requests in

response to receiving a second update request from the application prior to completing the first plurality of update requests as recited in claim 1. Thus, every feature of claim 1 is not expressly or inherently described in *Ohran*, *Armangau*, or *Ji*, either taken alone or in combination.

2. Order Dependency Based On Update Requests Is Not Disclosed or Suggested

Second, *Ohran*, *Armangau*, and *Ji*, either taken alone or in combination, fail to disclose or suggest that the first update request of the first plurality of update requests in the first group has an order dependency relative to the second update request of the second plurality of update requests in the second group. Claim 1 recites that there is an order dependency between two pluralities of update requests based on individual update requests within those plurality of update requests. The Examiner indicates that *Ohran* discloses that a first group T0-T1 has an order dependency relative to a second T1-T2 as shown in FIG. 2. However, Applicant respectfully submits that order dependency in *Ohran* is relative to time (for example, times T0, T1, T2, etc.), while order dependency in claim 1 is relative to receiving update requests (receiving the first update request and creating a group, then receiving the second update request prior to completion of the first group and creating the second group). *Armangau* fails to relieve this deficiency, as *Armangau* also fail to teach or suggest an order dependency between two groups of update requests based on individual update requests within those groups of update requests. *Ji* also fails to relieve this deficiency, as order dependency in *Ji* is also relative to time, while order dependency in claim 1 is relative to receiving update requests. Thus, every feature of claim 1 is not expressly or inherently described in *Ohran*, *Armangau*, or *Ji*, either taken alone or in combination.

3. Concurrently Completing Update Requests Is Not Disclosed or Suggested

Third, *Ohran*, *Armangau*, and *Ji*, either taken alone or in combination, fail to disclose or suggest concurrently completing the first plurality of update requests of the first group and, after concurrently completing the first plurality of update requests, concurrently completing the second plurality of update requests of the second group. Particularly, the Examiner alleges that "snapshot update processing for the first group could have been processed asynchronously according to figures 8A or 8B of *Armangau*." However, Appellants respectfully submit that FIG.

8A and FIG. 8B fail to disclose or suggest asynchronous processing at all, and at most disclose serial copying. In *Armangau*, snapshotted data merely has to be received and copied. However, when updating data includes sending update requests, the update requests must be received and the update request must be independently completed to update data at a backup system. Thus, as *Ohran* is wholly unconcerned with update requests, so too is *Armangau* wholly unconcerned with "completing" any update requests, as there are no update requests to complete.

Armangau also fails to disclose or suggest concurrently completing the update requests of the groups. Specifically, FIG. 8A and FIG. 8B illustrate serially copying snapshot data starting at the pointer for the first track and proceeding to the end of a production volume. *See Armangau* FIG. 8A blocks 131-135 and the iteration of blocks 132 to 135 to copy data, and FIG. 8B blocks 221-226 and the iteration of blocks 222 to 226 to send deallocation track pointers. Thus, *Armangau* merely discloses serially copying data as well as serially deallocating track pointers, and fails to disclose or suggest concurrently completing the first and second plurality of update requests. *Ohran* and *Ji* fail to relieve the deficiencies of *Armangau*, as *Ohran* and *Ji* similarly fail to disclose or suggest concurrently completing the first plurality of update requests of the first group, let alone concurrently completing the second plurality of update requests of the second group after concurrently completing the first plurality of update requests. Thus, every feature of claim 1 is not expressly or inherently described in *Ohran*, *Armangau*, or *Ji*, either taken alone or in combination.

4. Ohran Teaches Away From Sending Update Requests

Fourth, even assuming *arguendo* that the proposed combination disclosed every feature of claim 1, Appellants respectfully submit that there is no motivation to combine the references in the manner proposed by the Examiner because *Ohran* teaches away from claim 1. In particular, *Ohran* explicitly discloses that sending update requests is not preferred, as sending those update requests cause additional data and traffic to backup devices. See *Ohran*, col. 68, lines 31-51. As such, *Ohran* is explicitly, and exclusively, directed towards snapshotting only monitored physical sectors and is unconcerned with the operations that update those monitored physical sectors. *See Ohran*, col. 2 lines 17-38. Moreover, the Examiner admits that *Ohran* merely snapshots "changes made to a primary system over the course of time." *See* Final Office

Action, page 3 and the rejection of claim 1. Thus, *Ohran* would lead a person of ordinary skill in the art to follow a different path than creating groups of update requests. In re Gurley, 27 F.3d 551, 553, 31 USPQ2d 1130, 1131 (Fed. Cir. 1994) ("A reference may be said to teach away when a person of ordinary skill, upon reading the reference, would be discouraged from following the path set out in the reference, or would be led in a direction divergent from the path that was taken by the applicant."). Even KSR indicates that when the prior art teaches away from a claim, that claim is more likely to be nonobvious. KSR, 127 S. Ct. at 1739-1740. Therefore, the method of claim 1 that includes creating a first group including a first plurality of update requests in response to receiving a first update request from an application and creating a second group including a second plurality of update requests in response to receiving a second update request from the application prior to completing the first plurality of update requests is simply not the type of method contemplated by *Ohran*, and as such there is no objective reason to modify *Ohran* to incorporate the limitations recited by claim 1.

5. Ji Teaches Away From Creating Groups in Response to Update Requests

Fifth, Appellants respectfully submit that there is no motivation to combine the references in the manner proposed by the Examiner additionally because *Ji* teaches away from claim 1. In the most recent Office Action on page 3, the Examiner admits that *Ohran* fails to disclose or suggest creating a second group including a second plurality of update requests in response to receiving a second update request from the application prior to completing the first group of update requests. In an attempt to relieve the deficiencies of *Ohran*, the Examiner alleges that *Ji* may modify *Ohran* and disclose that limitation. As such, the Examiner alleges that *Ji* discloses creating a second group in response to receiving a second update request prior to completing the first plurality of update requests in *Ji*, paragraphs [0040] and [0042], as well as FIG. 3. *Ji*, however, merely discloses that write transactions are collected in batches, and the number of write transactions collected is based on batch size. *See Ji*, paragraph [0023]. Moreover, the Examiner clarifies that *Ohran* as modified by *Ji* would merely teach a method to "dynamically customize the size of group update packets to match the bandwidth between the primary storage and the secondary storage," and that such modification would have "reduced the

bandwidth requirements of the backup/snapshot system." However, the Examiner fails to clarify how *Ohran* as modified by *Ji* discloses or suggests creating a second group including a second plurality of update requests in response to receiving a second update request from the application prior to completing the first group of update requests. In fact, *Ji* would lead a person skilled in the art away from such a solution

Batch size in *Ji* may be determined based on the number of write transactions or size of write transactions (*Ji*, paragraph [0042]), based on the available bandwidth (*Ji*, paragraph [0046]), based on the bandwidth required to send a batch (*Ji*, paragraph [0048]), and based on the mean time between failures (*Ji*, paragraph [0049]). However, *Ji* teaches away from creating groups specifically in response to receiving update requests from an application. *Ji* discloses in paragraph [0043] that:

> If a write transaction received during the interval affects the same data as an earlier operation received during the same interval (and, thus, the later-received operation overwrites the prior data), the later-received operation may replace the earlier operation in the send batch.

Thus, *Ji* discloses that a first write transaction may be replaced by a second write transaction if data that is the target of both transactions is overwritten by the second write transaction. Thus, to the extent that the Examiner argues that *Ji* suggests creating a new group based upon an update request from the same application, *Ji* in fact teaches the opposite in attempting to combine update requests that are directed to the same data. *Ji* necessarily is prohibited from creating groups of update requests based on receiving update requests, as to do so would frustrate the purpose of allowing overwriting write transactions. Significantly, for this reason *Ji* discloses only creating batches of write transactions based solely on the size of the batches. *See Ji*, paragraph [0023]. *Ji* therefore would also lead a person of ordinary skill in the art to follow a different path than to create groups of update requests based on receiving update requests from an application. In re Gurley, 27 F.3d at 551. Moreover, Appellants respectfully submit that the Examiner's proposed modification would render *Ji* inoperable and/or change the principle of operation of the invention. McGinley v. Franklin Sports, Inc., 262 F.3d 1339, 1354, 60USPQ2d 1001, 1010 (a reference may teach away from a use when the use would render the result inoperable); In re Gordon, 733 F.3d 900, 902, 221 USPQ 1125, 1127 (Fed. Cir. 1984) (a

reference teaches away from proposed modifications that render the disclosed invention inoperable for its intended purpose); In re Ratti, 270 F.2d 810, 123 USPQ 349 (CCPA 1959) (if the proposed modification or combination of the prior art would change the principle of operation of the prior art invention being modified, then the teachings of the references are not sufficient to render the claims *prima facie* obvious). Therefore, the method of claim 1 that includes creating a first group including a first plurality of update requests in response to receiving a first update request from an application and creating a second group including a second plurality of update requests in response to receiving a second update request from the application prior to completing the first plurality of update requests is simply not the type of method contemplated by *Ji*, and as such there is no objective reason to modify *Ji* to incorporate the limitations recited by claim 1

6. The Examiner's Proposed Modifications Are Inharmonious

Finally, the Examiner's proposed modifications of *Ohran* would render it inoperable and/or change the principle of operation of the invention. The Examiner asserts that *Ohran* discloses creating a first group including a first plurality of update requests in response to receiving a first update request from an application. In support of this assertion, the Examiner cites FIG. 2 of *Ohran*. However, Appellants respectfully submit that the Examiner's proposed modifications would render *Ohran* inoperable and/or change the principle of operation of the invention.

As previously discussed, *Ohran* fails to disclose or suggest creating a first group including a plurality of update requests in response to a first update request from an application and *Ohran* further teaches away from sending update requests, relying instead on snapshotting of data at particular times. However, as *Ohran* teaches away from sending update requests, so too does *Ohran* teach away from sending batches of write transactions as disclosed in *Ji*. In short, the Examiner's proposed modification of *Ohran* to include sending groups or update requests or sending batches of write transactions would render *Ohran* inoperable and/or change the principle of operation of the invention.

With this in mind, Appellants respectfully submit that the Examiner has improperly shoehorned Appellants' invention into the disclosure of *Ohran*. Additionally, KSR lends no support to the Examiner's allegation on page 4 of the most recent Office Action that:

> It would have been obvious to one having ordinary skill in the art at the time the invention was made to have combined the backup system of modified *Ohran* with the teachings of using a predetermined size for the batch updates instead of a time interval (e.g., the T_0 to T_1 of *Ohran*) in order to have been able to dynamically customize the size of group update packets to match the bandwidth between the primary storage and secondary storage of modified *Ohran*.

KSR does not suggest that obviousness can be established by combining disparate and mutually exclusive teachings. In fact, Appellants respectfully submit that KSR cautions against the type of *ex post* reasoning the Examiner applies in their Office Action. KSR, 127 S. Ct. at 1729 (citing Graham v. John Deere, 383 U.S. at 36) ("A factfinder should be aware, of course, of the distortion caused by hindsight bias and must be cautious of arguments reliant on *ex post* reasoning."). Thus, Appellants respectfully submit that the Examiner's combination of *Ohran*, *Armangau*, and *Ji* create an unworkable amalgamation of disparate teachings, and thus Appellants respectfully submit that the Examiner's rejection of claim 1 is improper.

Appellants therefore respectfully submit that the Examiner's proposed combination of *Ohran, Armangau* and *Ji* would not make the claimed invention "obvious…to a person having ordinary skill in the art." 35 U.S.C. §103(a). Appellants respectfully submit that the Examiner has failed to establish a *prima facie* case of obviousness as to claim 1 and that the Examiner's rejections are clear error and should be reversed. Reversal of the Examiner's rejection of claim 1, and allowance of that claim, and of claims 3, 4, 6-13 and 36 which depend therefrom, are therefore respectfully requested.

Independent Claim 14

Next, with regard to the rejection of independent claim 14, this claim generally recites a method for updating data at a backup system that tracks updates made to a primary system. The method includes synchronously processing a plurality of groups of update requests, a first update request from an application in a first group of update requests from among the plurality of groups

having an order dependency relative to a second update request from the application in a second group of update requests from among the plurality of groups, with the update requests in each group being capable of being processed concurrently and without regard for order relative to one another, and wherein receipt of the second update request prior to processing of the first update request initiates the creation of the second group of update requests. The method further includes asynchronously processing the update requests in each group.

As discussed above in connection with claim 1, the cited art, whether taken alone or in combination , fails to disclose or suggest that receipt of the second update request prior to processing of the first update request initiates the creation of the second group of update requests. In addition, both *Ohran* and *Ji* teach away from the invention recited in claim 14, and the Examiner's proposed modifications of *Ohran* would render it inoperable and/or change the principle of operation of the invention.

Furthermore, claim 14 additionally recites that asynchronously processing the update requests in each group. In rejecting claim 14, the Examiner relies on a combination of *Ohran, Armangau* and *Ji*. The Examiner follows the rejection for claim 1 above, and the Examiner alleges that *Ohran* discloses that the groups of updates are processed synchronously (according to a time when the updates are received, or alternatively by the amount of data contained in the batch update groups in paragraphs 40-42 of *Ji*). The Examiner further alleges that *Armangau* teaches that the updates of each group can be performed asynchronously during snapshot processing in col. 17, lines 6-28. Similar to the discussion above in connection with claim 1, *Armangau* fails to disclose or suggest asynchronously performing the updates of each group, and at most discloses serial copying. In *Armangau*, snapshotted data merely has to be received and copied. However, as *Ohran* is wholly unconcerned with update requests, so too is *Armangau* wholly unconcerned with "performing" any update requests, as there are no update requests to complete.

Moreover, and also similar to the discussion above in connection with claim 1, *Armangau* also fails to disclose or suggest asynchronously performing the updates of each group. Specifically, FIG. 8A and FIG. 8B illustrate serially copying snapshot data starting at the pointer for the first track and proceeding to the end of a production volume. *See Armangau* FIG. 8A

blocks 131-135 and the iteration of blocks 132 to 135 to copy data, and FIG. 8B blocks 221-226 and the iteration of blocks 222 to 226 to send deallocation track pointers. Thus, *Armangau* merely discloses serially copying data as well as serially deallocating track pointers, and fails to disclose or suggest asynchronously performing the updates of each group. *Ohran* and *Ji* fail to relieve the deficiencies of *Armangau*, as *Ohran* and *Ji* similarly fail to disclose or suggest asynchronously performing the updates of each group. Thus, every feature of claim 14 is not expressly or inherently described in *Ohran*, *Armangau*, or *Ji*, either taken alone or in combination.

Applicants therefore respectfully submit that the Examiner's proposed combination of *Ohran, Armangau* and *Ji* would not make the claimed invention "obvious…to a person having ordinary skill in the art." 35 U.S.C. §103(a). Appellants respectfully submit that the Examiner has failed to establish a *prima facie* case of obviousness as to claim 14 and that the Examiner's rejections are clear error and should be reversed. Reversal of the Examiner's rejection of claim 14, and allowance of that claim, and of claims 15 and 16 which depend therefrom, are therefore respectfully requested

Dependent Claim 6

Dependent claim 6 depends from independent claim 1 and further recites that creating the first group further includes updating a status indicative of whether the first group is active. In rejecting claim 6, the Examiner alleges that this claim is obvious in light of *Ohran*, *Armangau*, and *Ji*, and further refers to *Ohran* and col. 9 line 63, to col. 10 line 8. In particular, this passage discloses:

> In FIG. 3, the means for preserving a static snapshot is illustrated by snapshot processing block 50. As illustrated in FIG. 3, it may make sense to incorporate the snapshot processing mechanism into the mass storage read/write processing block. Although the details of snapshot processing block 50 are represented below, one preferred embodiments preserves a static snapshot by copying a data block of the original data 14 that is to be overwritten from the original data 14 into snapshot storage 22 and then indicating in snapshot map 52 that the block has been preserved in snapshot storage 22. Once a copy has been placed into snapshot storage 22, then the copy of the data block in the original data 14 can be overwritten.

Thus the Examiner is considering that when a snapshot bitmap contains any data the current group may be considered "active" as the setting of a bit in the bitmap would indicate the presence of a new update. Appellants respectfully submit that this consideration is improper and the rejection of claim 6 should be reversed.

In rejecting claim 6, the Examiner reads limitations into claim 6 that are not recited in that claim. Claim 6 recites that "creating the first group further includes updating a status indicative of whether the first group is active." The Examiner alleges that storing any data into a snapshot bitmap is analogous to "updating a status indicative of whether the first group is active." To the extent the Examiner maintains this rejection is clear error. The storing of data in a snapshot bitmap is a separate action from updating a status indicative of whether the first group is active, and updating the status may be performed independently of storing data in a snapshot bitmap, as the status and the data of the bitmap may be two separate and independent entities. As such, Appellants respectfully submit that the Examiner has improperly read limitations into claim 6 that are not in that claim and that the Examiner's rejection is clear error and should be reversed. *Armangau* and *Ji*, as well as the other cited art, fail to relieve the deficiencies of *Ohran*, as those references are not directed towards updating the status of a group of update requests. Thus, every feature of claim 6 is not expressly or inherently described in *Ohran*, *Armangau*, or *Ji*, either taken alone or in combination. Appellants therefore submit that claim 6 is novel and non-obvious over the prior art of record. Reversal of the Examiner's rejection of claim 6 and allowance of that claim are therefore respectfully requested.

Dependent Claims 8, 10-13, and 15

Dependent claims 8, 10-13, and 15 are not argued separately.

Dependent Claim 36

Dependent claim 36 depends from independent claim 1 and further recites, after completing the first plurality of update requests, arranging the second plurality of update requests according to the order dependency. In rejecting claim 36, the Examiner alleges that this claim is obvious in light of *Ohran*, *Armangau*, and *Ji*, and further refers to *Ohran* and FIG. 2. In particular, the Examiner alleges, that *Ohran*, "by default, teaches arranging the second group of

requests (e.g., updates occurring to the original data from time T1-T2) as all of the second group of requests are arranged by order dependency to be processed after the first plurality of requests as shown in the timeline of figure 2." However, *Ohran*, as admitted by the Examiner, merely makes a "collection of all changes made to the primary system from T0-T1," and, as previously discussed, fails to disclose or suggest collecting update requests at all. Thus, *Ohran* must necessarily fail to disclose or suggest arranging update requests if it fails to disclose or suggest update requests. *Armangau* fails to relieve the deficiencies of *Ohran*, as *Armangau* is directed towards buffering snapshots and thus also fails to disclose or suggest arranging the second plurality of update requests according to the order dependency after completing the first plurality of update requests. Moreover, *Ji* also fails to relieve the deficiencies of *Ohran* and *Armangau*, as Ji also fails to disclose or suggest arranging the second plurality of update requests according to the order dependency after completing the first plurality of update requests, and as Ji teaches away from that limitation by disclosing that overwrites may be allowed across batch barriers, thus frustrating the purpose of arranging the second plurality of update requests. *See Ji*, paragraph [0043]. Thus, every feature of claim 36 is not expressly or inherently described in *Ohran*, *Armangau*, or *Ji*, either taken alone or in combination. Appellants therefore submit that claim 36 is novel and non-obvious over the prior art of record. Reversal of the Examiner's rejection of claim 36 and allowance of that claim are therefore respectfully requested.

B. Claim 3 is Patentable Over *Ohran* In View of *Armangau*, *Ji*, and In Further View of *Yamagami*

Dependent claim 3 depends from independent claim 1and further recites that creating the first group further includes creating a group that includes a plurality of requests initiated at a plurality of applications. In rejecting claim 3, the Examiner admits that this feature is not disclosed by *Ohran, Armangau*, and *Ji*, but relies on *Yamagami* for allegedly disclosing this additional feature. However, irrespective of whether *Yamagami* discloses the additional feature recited in claim 3, *Yamagami* does not address the aforementioned deficiencies in the other references with respect to the parent claim, independent claim 1. Thus, Appellants respectfully submit that claim 3 is at least patentable by virtue of its dependency on the aforementioned independent claim 1.

C. Claim 4 is Patentable Over *Ohran* In View of *Armangau*, *Ji*, and In Further View of *Zait*

Dependent claim 4 depends from independent claim 1and further recites that creating the first group further includes updating a count associated with a number of the first plurality of update requests. In rejecting claim 4, the Examiner admits that this feature is not disclosed by *Ohran, Armangau,* and *Ji,* but relies on *Zait* for allegedly disclosing this additional feature. However, irrespective of whether *Zait* discloses the additional feature recited in claim 4, *Zait* does not address the aforementioned deficiencies in the other references with respect to the parent claim, independent claim 1. Thus, Appellants respectfully submit that claim 4 is at least patentable by virtue of its dependency on the aforementioned independent claim 1.

D. Claims 7 and 16 are Patentable Over *Ohran* In View of *Armangau*, *Ji*, and In Further View of *Kapoor*

Dependent claims 7 and 16 depend from respective independent claims 1 and 14 and further recite assigning a group number to each update request of the first plurality of update requests. In rejecting claims 7 and 16, the Examiner admits that the additional feature in claims 7 and 16 is not disclosed by *Ohran, Armangau,* and *Ji,* but relies on *Kapoor* for allegedly disclosing this additional feature. However, irrespective of whether *Kapoor* discloses the additional feature recited in claims 7 and 16, *Kapoor* does not address the aforementioned deficiencies in the other references with respect to the parent claims, independent claim 1 and independent claim 14, respectively. Thus, Appellants respectfully submit that claim 7 is at least patentable by virtue of its dependency on the aforementioned independent claim 1. Additionally, Appellants respectfully submit that claim 16 is at least patentable by virtue of its dependency on the aforementioned claim 14.

E. Claim 9 is Patentable Over *Ohran* In View of *Armangau*, *Ji*, and In Further View of *Golds*

Dependent claim 9 depends from independent claim 1and further recites that creating the first group further includes reading a group number from an update request. In rejecting claim 9, the Examiner admits that this additional feature is not disclosed in *Ohran, Armangau,* and *Ji,* but

relies on *Golds* for allegedly disclosing this additional featuer. However, irrespective of whether *Golds* discloses the additional feature recited in claim 9, *Golds* does not address the aforementioned deficiencies in the other references with respect to the parent claim, independent claim 1. Thus, Appellants respectfully submit that claim 9 is at least patentable by virtue of its dependency on the aforementioned independent claim 1.

F. Conclusion

Appellants respectfully requests that the Board reverse the Examiner's rejections of claims 1, 3, 4, 6-16 and 36, and that the Application be passed to issue. If there are any questions regarding the foregoing, please contact the undersigned at (513) 241-2324. The Appeal Brief Fee is being submitted concurrently herewith. If any other charges or credits are necessary to complete this communication, please apply them to Deposit Account 23-3000.

Respectfully submitted,

November 24, 2008
Date

/Scott A. Stinebruner/
Scott A. Stinebruner
Reg. No. 38,323
WOOD, HERRON & EVANS, L.L.P.
2700 Carew Tower
441 Vine Street
Cincinnati, Ohio 45202
Telephone: (513) 241-2324
Facsimile: (513) 241-6234

VIII. CLAIMS APPENDIX: CLAIMS ON APPEAL (S/N 10/758,484)

Listing of Claims:

1. (Previously Presented) A method for updating data at a backup system that tracks updates made to a primary system, the method comprising:

in response to receiving a first update request from an application, creating a first group including a first plurality of update requests, the first plurality of update requests including the first update request;

in response to receiving a second update request from the application prior to completing the first plurality of update requests, creating a second group including a second plurality of update requests, the second plurality of update requests including the second update request, the first update request of the first plurality of update requests in the first group having an order dependency relative to the second update request of the second plurality of update requests in the second group, with the update requests in each of the first and second groups capable of being processed concurrently and without regard for order relative to one another;

concurrently completing the first plurality of update requests of the first group; and

after concurrently completing the first plurality of update requests, concurrently completing the second plurality of update requests of the second group.

2. (Canceled)

3. (Previously Presented) The method of claim 1, wherein creating the first group further includes creating a group that includes a plurality of requests initiated at a plurality of applications.

4. (Previously Presented) The method of claim 1, wherein creating the first group further includes updating a count associated with a number of the first plurality of update requests.

5. (Canceled)

6. (Previously Presented) The method of claim 1, wherein creating the first group further includes updating a status indicative of whether the first group is active.

7. (Previously Presented) The method of claim 1, wherein creating the first group further includes assigning a group number to each update request of the first plurality of update requests.

8. (Previously Presented) The method of claim 1, wherein concurrently completing the first plurality of update requests further includes issuing an update request of the first plurality of update requests.

9. (Previously Presented) The method of claim 1, wherein creating the first group further includes reading a group number from an update request.

10. (Previously Presented) The method of claim 1, wherein concurrently completing the first plurality of update requests further includes holding the second-update request.

11. (Previously Presented) The method of claim 10, wherein concurrently completing the second plurality of update requests further includes releasing a hold on the second update request.

12. (Previously Presented) The method of claim 1, wherein creating the first group, creating the second group, concurrently completing the first plurality of update requests and concurrently completing the second plurality of update requests further comprises creating the first group, creating the second group, completing the first plurality of update requests and completing the second plurality of update requests on the primary system.

13. (Previously Presented) The method of claim 1, wherein creating the first group, creating the second group, completing the first plurality of update requests and completing the

second plurality of update requests further comprises creating the first group, creating the second group, completing the first plurality of update requests and completing the second plurality of update requests on the backup system.

14. (Previously Presented) A method for updating data at a backup system that tracks updates made to a primary system, the method comprising:

synchronously processing a plurality of groups of update requests, a first update request from an application in a first group of update requests from among the plurality of groups having an order dependency relative to a second update request from the application in a second group of update requests from among the plurality of groups, with the update requests in each group being capable of being processed concurrently and without regard for order relative to one another, and wherein receipt of the second update request prior to processing of the first update request initiates the creation of the second group of update requests; and

asynchronously processing the update requests in each group.

15. (Previously Presented) The method of claim 14, wherein processing the plurality of update requests further includes holding the second update request of the second group from among the plurality of groups.

16. (Original) The method of claim 14, wherein processing the groups further includes assigning a group number to an update request of the plurality of update requests.

17. – 35. (Canceled)

36. (Previously Presented) The method of claim 1, further comprising after completing the first plurality of update requests, arranging the second plurality of update requests according to the order dependency.

37. (Canceled)

IX. EVIDENCE APPENDIX (S/N 10/758,484)

None.

X. RELATED PROCEEDINGS APPENDIX (S/N 10/758,484)

None.